In [2]:
```
import pandas as pd

# Load the dataset
real_estate_data = pd.read_csv("Real_Estate.csv")

# Display the first few rows of the dataset and the info about the dataset
real_estate_data_head = real_estate_data.head()
data_info = real_estate_data.info()

print(real_estate_data_head)
print(data_info)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 414 entries, 0 to 413
Data columns (total 7 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Transaction date                        414 non-null    object
 1   House age                               414 non-null    float64
 2   Distance to the nearest MRT station     414 non-null    float64
 3   Number of convenience stores            414 non-null    int64
 4   Latitude                                414 non-null    float64
 5   Longitude                               414 non-null    float64
 6   House price of unit area                414 non-null    float64
dtypes: float64(5), int64(1), object(1)
memory usage: 22.8+ KB
None
   Transaction date  House age  Distance to the nearest MRT station  \
0  2012-09-02 16:42:30.519336       13.3                               4082.0150
1  2012-09-04 22:52:29.919544       35.5                                274.0144
2  2012-09-05 01:10:52.349449        1.1                               1978.6710
3  2012-09-05 13:26:01.189083       22.2                               1055.0670
4  2012-09-06 08:29:47.910523        8.5                                967.4000

   Number of convenience stores   Latitude   Longitude  \
0                             8  25.007059  121.561694
1                             2  25.012148  121.546990
2                            10  25.003850  121.528336
3                             5  24.962887  121.482178
4                             6  25.011037  121.479946

   House price of unit area
0                  6.488673
1                 24.970725
2                 26.694267
3                 38.091638
4                 21.654710
None
```

In [3]:
```
print(real_estate_data.isnull().sum())
```

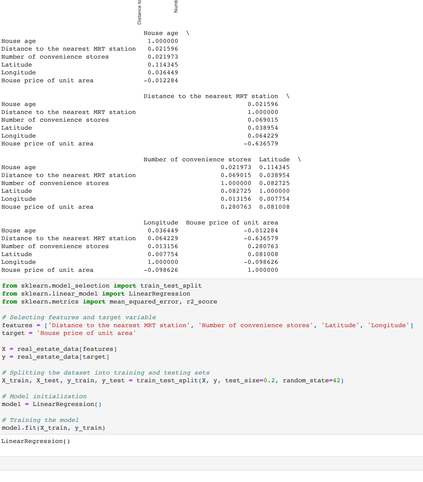
```
Transaction date                        0
House age                               0
Distance to the nearest MRT station     0
Number of convenience stores            0
Latitude                                0
Longitude                               0
House price of unit area                0
dtype: int64
```

In [4]:
```
descriptive_stats = real_estate_data.describe()

print(descriptive_stats)
```

```
        House age  Distance to the nearest MRT station  \
count  414.000000                          414.000000
mean    18.405072                         1064.468233
std     11.757670                         1196.749385
min      0.000000                           23.382840
25%      9.900000                          289.324800
50%     16.450000                          506.114400
75%     30.375000                         1454.279000
max     42.700000                         6306.153000

       Number of convenience stores    Latitude   Longitude  \
count                    414.000000  414.000000  414.000000
mean                       4.265700   24.973605  121.520268
std                        2.880498    0.024178    0.026989
min                        0.000000   24.932075  121.473888
25%                        2.000000   24.952422  121.496866
50%                        5.000000   24.974353  121.520912
75%                        6.750000   24.994947  121.544676
max                       10.000000   25.014578  121.565321

       House price of unit area
count                414.000000
mean                  29.102149
std                   15.750935
min                    0.000000
25%                   18.422493
50%                   30.394070
75%                   40.615184
max                   65.571716
```

In [5]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Set the aesthetic style of the plots
sns.set_style('whitegrid')

# Create histograms for the numerical columns
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(12, 12))
fig.suptitle('Histograms of Real Estate Data', fontsize=16)

cols = ['House age', 'Distance to the nearest MRT station', 'Number of convenience stores',
        'Latitude', 'Longitude', 'House price of unit area']

for i, col in enumerate(cols):
    sns.histplot(real_estate_data[col], kde=True, ax=axes[i//2, i%2])
    axes[i//2, i%2].set_title(col)
    axes[i//2, i%2].set_xlabel('')
    axes[i//2, i%2].set_ylabel('')

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

In [6]:
```
# Scatter plots to observe the relationship with house price
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 10))
fig.suptitle('Scatter Plots with House Price of Unit Area', fontsize=16)

# Scatter plot for each variable against the house price
sns.scatterplot(data=real_estate_data, x='House age', y='House price of unit area', ax=axes[0, 0])
sns.scatterplot(data=real_estate_data, x='Distance to the nearest MRT station', y='House price of unit area', ax=axes[0, 1])
sns.scatterplot(data=real_estate_data, x='Number of convenience stores', y='House price of unit area', ax=axes[1, 0])
sns.scatterplot(data=real_estate_data, x='Latitude', y='House price of unit area', ax=axes[1, 1])

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```



In [7]:
```
# Correlation matrix
correlation_matrix = real_estate_data.corr()

# Plotting the correlation matrix
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Matrix')
plt.show()

print(correlation_matrix)
```



```
                                      House age  \
House age                              1.000000
Distance to the nearest MRT station    0.021596
Number of convenience stores           0.021973
Latitude                               0.114345
Longitude                              0.036449
House price of unit area              -0.012284

                                      Distance to the nearest MRT station  \
House age                                                        0.021596
Distance to the nearest MRT station                              1.000000
Number of convenience stores                                     0.069015
Latitude                                                         0.038954
Longitude                                                        0.064229
House price of unit area                                        -0.636579

                                      Number of convenience stores  Latitude  \
House age                                                 0.021973  0.114345
Distance to the nearest MRT station                       0.069015  0.038954
Number of convenience stores                              1.000000  0.083725
Latitude                                                  0.083725  1.000000
Longitude                                                 0.013156  0.007754
House price of unit area                                  0.280763  0.081008

                                      Longitude  House price of unit area
House age                              0.036449                 -0.012284
Distance to the nearest MRT station    0.064229                 -0.636579
Number of convenience stores           0.013156                  0.280763
Latitude                               0.007754                  0.081008
Longitude                              1.000000                 -0.098626
House price of unit area              -0.098626                  1.000000
```

In [8]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Selecting features and target variable
features = ['Distance to the nearest MRT station', 'Number of convenience stores', 'Latitude', 'Longitude']
target = 'House price of unit area'

X = real_estate_data[features]
y = real_estate_data[target]

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Model initialization
model = LinearRegression()

# Training the model
model.fit(X_train, y_train)
```

Out[8]:
```
LinearRegression()
```

In [ ]: